HW1
Food Multi-classification

Outline

Objection

Task Descriptin

Data

Kaggle

Grading

Deadlines

Report

Hints

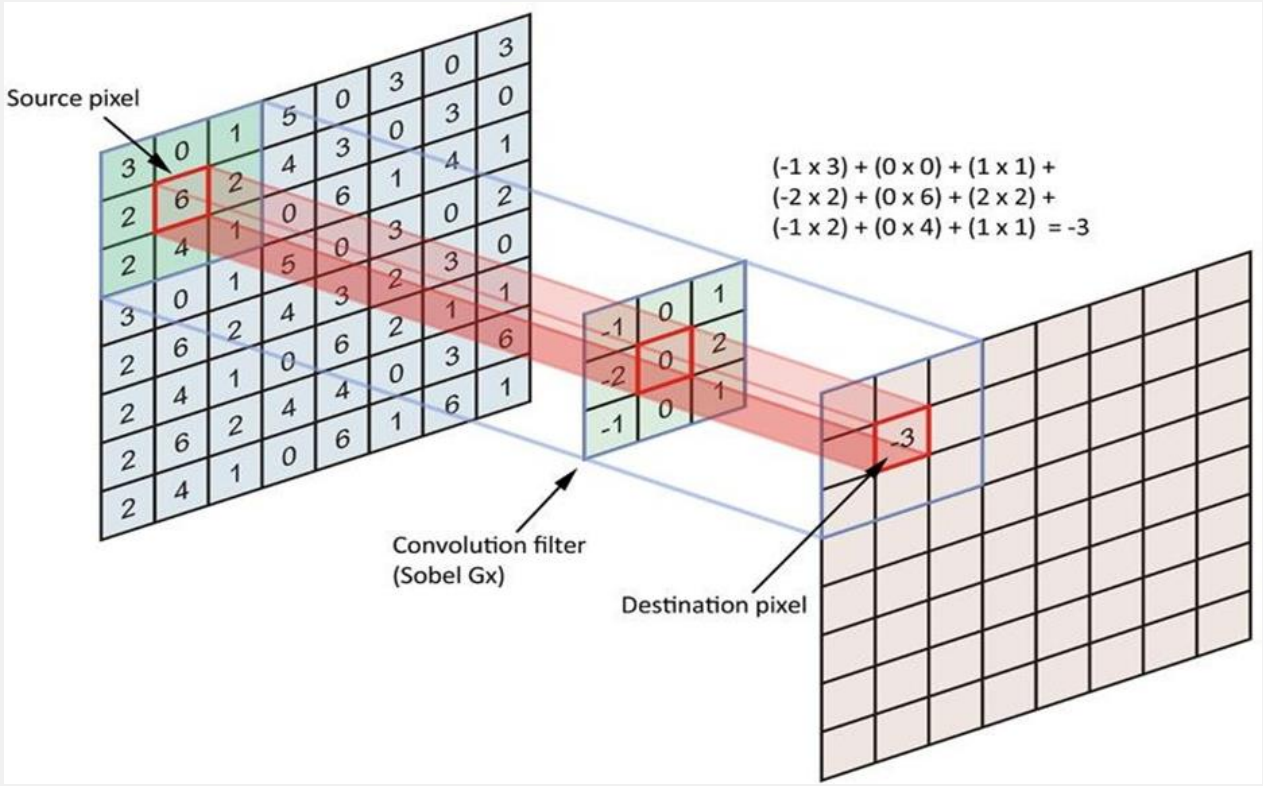
Regulations

Objection

Using CNN for multi-classification

Using data augmentation to improve CNN performance

Adjusting depth of CNN for different case difficulty



Task Description

Categorize 11 different food images

Data

Categorize 11 different food images

Training data:

- 9866 images consist of 11 categories

Test data:

- 3430 images

Data

Training directory

Test directory

Kaggle

Join Link: Kaggle

Sample Code: Code

Dataset: Data

Name setting: <student ID>_<anything>

- E.g. M1254xxx_恐龍抗郎抗郎抗, M1254xxx_wakuwaku

Submission format: *.csv file

Id	Category
0	5
1	1
2	9
3	9
4	8
5	1
6	6
7	0
8	6

Dataset Loader


```python
class FoodDataset(Dataset):

    def __init__(self,path,tfm=test_tfm,files = None):
        super(FoodDataset).__init__()
        self.path = path
        self.files = sorted([os.path.join(path,x) for x in os.listdir(path) if x.endswith(".jpg")])
        if files != None:
            self.files = files
        print(f"One {path} sample",self.files[0])
        self.transform = tfm

    def __len__(self):
        return len(self.files)

    def __getitem__(self,idx):
        fname = self.files[idx]
        im = Image.open(fname)
        im = self.transform(im)
        #im = self.data[idx]
        try:
            label = int(fname.split("/")[-1].split("_")[0])
        except:
            label = -1 # test has no label
        return im,label
```

Model Construction

Model Construction

```
class  Classifier(nn.Module):
        def   __init__(self):
                super(Classifier,   self).__init__()
                pass

        def   forward(self,   x):
                pass
```

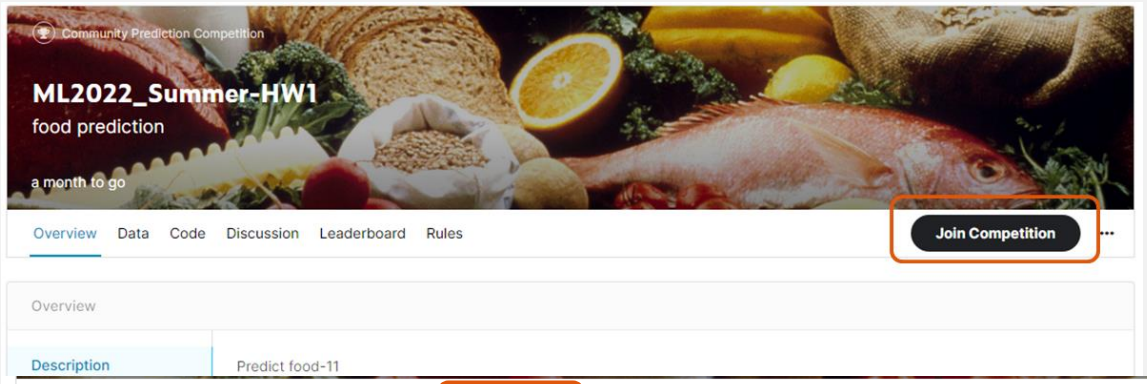
Test and Generate prediction CSV

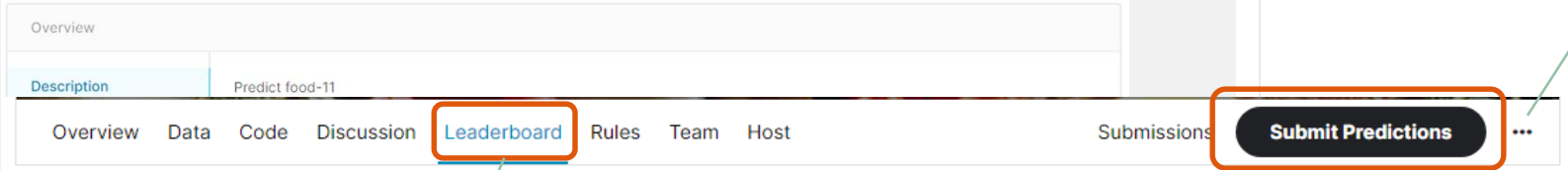
Testing and generate prediction CSV

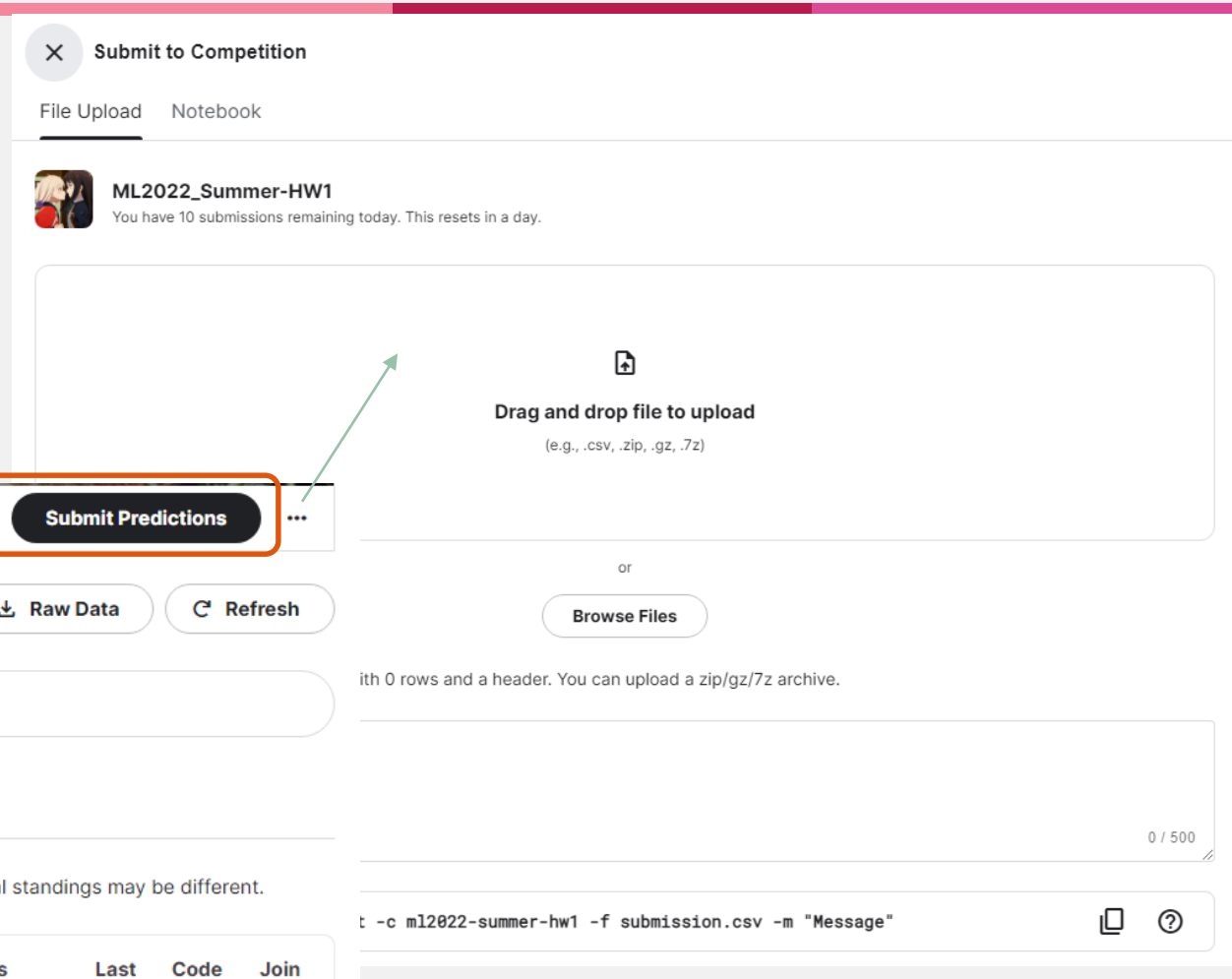
```
[ ]  model_best = Classifier().to(device)
     model_best.load_state_dict(torch.load(f"{_exp_name}_best.ckpt"))
     model_best.eval()
     prediction = []
     with torch.no_grad():
             for data,_ in test_loader:
                     test_pred = model_best(data.to(device))
                     test_label = np.argmax(test_pred.cpu().data.numpy(), axis=1)
                     prediction += test_label.squeeze().tolist()
```

```
[ ]  #create test csv
     def pad4(i):
             return "0"*(4-len(str(i)))+str(i)
     df = pd.DataFrame()
     df["Id"] = [pad4(i) for i in range(1,len(test_set)+1)]
     df["Category"] = prediction
     df.to_csv("submission.csv",index = False)
```

Kaggle







Leaderboard

Raw Data Refresh

Search leaderboard

Public Private

This leaderboard is calculated with approximately 50% of the test data. The final results will be based on the other 50%, so the final standings may be different.

#	Team	Members	Score	Entries	Last	Code	Join
	=====I'm very strong======		0.67463				
	=====Baseline=====		0.53994				

Kaggle

Maximum Daily Submission: 10 (UTC+8)(8:00 am)

Select 2 submission for private leaderboard (在Deadline之前選擇，Deadline之後公布，記得要選，沒選以0分計算)

Grading

Baseline (Public)	- 15pts
Baseline (Private)	- 15pts
Strong baseline (Public)	- 15pts
Strong baseline (Private)	- 15pts
Report (Public)	- 40pts

會根據同學繳交情況調整

Deadlines

Kaggle and Report

- 2023/11/08 23:59 (UTC+8) Upload to NCUE Cloud Server
- 遲交分數會打折

Report Format

Architecture

Method (with code highline)

Learned with course and homework

Conclusion

Hints

Using good famous CNN architecture

Pretrained model (Prohibit)

Augmentation

Regulations

請自己完成作業，勿分享code與參數

禁止手動更改預測結果

禁止使用任何方法在一天內提交超過所限制的次數